Terms used **reference counting collector** Found **33,717** of **164,603**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 An on-the-fly reference counting garbage collector for Java

Yossi Levanoni, Erez Petrank

October 2001 **ACM SIGPLAN Notices , Proceedings of the 16th ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications**, Volume 36 Issue 11

Full text available: pdf(280.30 KB) Additional Information: full citation, abstract, references, citings, index terms

Reference counting is not naturally suitable for running on multiprocessors. The update of pointers and reference counts requires atomic and synchronized operations. We present a novel reference counting algorithm suitable for a multiprocessor that does not require any synchronized operation in its write barrier (not even a compare-and-swap type of synchronization). The algorithm is efficient and may complete with any tracing algorithm.

2 Ulterior reference counting: fast garbage collection without a long wait

Stephen M. Blackburn, Kathryn S. McKinley

October 2003 **ACM SIGPLAN Notices , Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications**, Volume 38 Issue 11

Full text available: pdf(218.61 KB) Additional Information: full citation, abstract, references, citings, index terms

General purpose garbage collectors have yet to combine short pause times with high throughput. For example, generational collectors can achieve high throughput. They have modest average pause times, but occasionally collect the whole heap and consequently incur long pauses. At the other extreme, concurrent collectors, including reference counting, attain short pause times but with significant performance penalties. This paper introduces a new hybrid collector that combines copying generational c ...

Keywords: Java, copying, generational hybrid, reference counting, ulterior reference counting

3 The space cost of lazy reference counting

Hans-J. Boehm

January 2004 **ACM SIGPLAN Notices , Proceedings of the 31st ACM SIGPLAN-SIGACT symposium on Principles of programming languages**, Volume 39 Issue 1

Full text available: pdf(125.88 KB) Additional Information: full citation, abstract, references, index terms

Reference counting memory management is often advocated as a technique for reducing or avoiding the pauses associated with tracing garbage collection. We present some measurements to remind the reader that classic reference count implementations may in fact exhibit longer pauses than tracing collectors.We then analyze reference counting with lazy deletion, the standard technique for avoiding long pauses by deferring deletions and associated reference count decrements, usually to allocation time. ...

Keywords: garbage collection, memory allocation, reference counting, space complexity

4 Lock-free reference counting

David L. Detlefs, Paul A. Martin, Mark Moir, Guy L. Steele

August 2001 **Proceedings of the twentieth annual ACM symposium on Principles of distributed computing**

Full text available: pdf(802.52 KB) Additional Information: full citation, abstract, references, citings, index terms

Assuming the existence of garbage collection makes it easier to design implementations of concurrent data structures. However, this assumption limits their applicability. We present a methodology that, for a significant class of data structures, allows designers to first tackle the easier problem of designing a garbage-collection-dependent implementation, and then apply our methodology to achieve a garbage-collection-independent one. Our methodology is based on the well-known reference counti ...

5 Garbage collection in object-oriented databases using transactional cyclic reference counting

P. Roy, S. Seshadri, A. Silberschatz, S. Sudarshan, S. Ashwin

August 1998 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 7 Issue 3

Full text available: pdf(180.00 KB) Additional Information: full citation, abstract, index terms

Garbage collection is important in object-oriented databases to free the programmer from explicitly deallocating memory. In this paper, we present a garbage collection algorithm, called Transactional Cyclic Reference Counting (TCRC), for object-oriented databases. The algorithm is based on a variant of a reference-counting algorithm proposed for functional programming languages The algorithm keeps track of auxiliary reference count information to detect and collect cyclic garbage. The algorithm ...

6 Morris's Garbage Compaction Algorithm Restores Reference Counts

David S. Wise

January 1979 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 1 Issue 1

Full text available: pdf(303.55 KB) Additional Information: full citation, abstract, references, citings, index terms

The two-pass compaction algorithm of F.L. Morris, which follows upon the mark phase in a garbage collector, may be modified to recover reference counts for a hybrid storage management system. By counting the executions of two loops in that algorithm where upward and downward references, respectively, are forwarded to the relocation address of one node, we can initialize a count of active references and then update it but once. The reference count may share space with the mark bit in each no ...

7 Minimizing reference count updating with deferred and anchored pointers for functional data structures

Henry G. Baker

September 1994 **ACM SIGPLAN Notices**, Volume 29 Issue 9

Full text available: Additional Information:

pdf(664.75 KB) full citation, citings, index terms

8 Java without the coffee breaks: a nonintrusive multiprocessor garbage collector
David F. Bacon, Clement R. Attanasio, Han B. Lee, V. T. Rajan, Stephen Smith
May 2001 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2001 conference on Programming language design and implementation**, Volume 36 Issue 5
Full text available: pdf(1.69 MB) Additional Information: full citation, abstract, references, citings, index terms

The deployment of Java as a concurrent programming language has created a critical need for high-performance, concurrent, and incremental multiprocessor garbage collection. We present the *Recycler*, a fully concurrent pure reference counting garbage collector that we have implemented in the Jalapeño Java virtual machine running on shared memory multiprocessors.

While a variety of multiprocessor collectors have been proposed and some have been implemented, experimental dat ...

9 Hierarchical distributed reference counting
Luc Moreau
October 1998 **ACM SIGPLAN Notices , Proceedings of the 1st international symposium on Memory management**, Volume 34 Issue 3
Full text available: pdf(1.12 MB) Additional Information: full citation, abstract, references, citings, index terms

Massively distributed computing is a challenging problem for garbage collection algorithm designers as it raises the issue of scalability. The high number of hosts involved in a computation can require large tables for reference listing, whereas the lack of information sharing between hosts in a same locality can entail redundant GC traffic. In this paper, we argue that a conceptual hierarchical organisation of massive distributed computations can solve this problem. By conceptual hierarchical o ...

10 Diffusion tree restructuring for indirect reference counting
Peter Dickman
October 2000 **ACM SIGPLAN Notices , Proceedings of the 2nd international symposium on Memory management**, Volume 36 Issue 1
Full text available: pdf(1.32 MB) Additional Information: full citation, abstract, citings, index terms

A new variant algorithm for distributed acyclic garbage detection is presented for use in hybrid garbage collectors. The existing fault-tolerance of Piquer's Indirect Reference Counting (IRC) is qualitatively improved by this new approach. The key insight that underpins this work is the observation that the parent of a node in the IRC diffusion tree need not remain constant. The new variant exploits standard mechanisms for implementing diffusion trees and remote references, using four simple ...

11 A distributed garbage collector with diffusion tree reorganisation and mobile objects
Luc Moreau
September 1998 **ACM SIGPLAN Notices , Proceedings of the third ACM SIGPLAN international conference on Functional programming**, Volume 34 Issue 1
Full text available: pdf(1.44 MB) Additional Information: full citation, abstract, references, citings, index terms

We present a new distributed garbage collection algorithm that is able to reorganise diffusion trees and to support mobile objects. It has a modular design comprising three components: a reliable transport mechanism, a reference-counting based distributed garbage collector for non-mobile objects, and an extra layer that provides mobility. The

algorithm is formalised by an abstract machine and is proved to be correct. The safety property ensures that an object may not be reclaimed as long as it i ...

12 An on-the-fly mark and sweep garbage collector based on sliding views
Hezi Azatchi, Yossi Levanoni, Harel Paz, Erez Petrank
October 2003 **ACM SIGPLAN Notices , Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications**, Volume 38 Issue 11

Full text available: pdf(244.12 KB) Additional Information: full citation, abstract, references, citings, index terms

With concurrent and garbage collected languages like Java and C# becoming popular, the need for a suitable non-intrusive, efficient, and concurrent multiprocessor garbage collector has become acute. We propose a novel mark and sweep on-the-fly algorithm based on the sliding views mechanism of Levanoni and Petrank. We have implemented our collector on the Jikes Java Virtual Machine running on a Netfinity multiprocessor and compared it to the concurrent algorithm and to the stop-the-world collecto ...

Keywords: concurrent garbage collection, garbage collection, memory management, on-the-fly garbage collection, runtime systems

13 Robust, distributed references and acyclic garbage collection
Marc Shapiro, Peter Dickman, David Plainfossé
October 1992 **Proceedings of the eleventh annual ACM symposium on Principles of distributed computing**

Full text available: pdf(1.27 MB) Additional Information: full citation, references, citings, index terms, review

14 Managing Reentrant Structures Using Reference Counts
Daniel G. Bobrow
July 1980 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 2 Issue 3

Full text available: pdf(317.55 KB) Additional Information: full citation, abstract, references, citings, index terms

Automatic storage management requires that one identify storage unreachable by a user's program and return it to free status. One technique maintains a count of the references from user's programs to each cell, since a count of zero implies the storage is unreachable. Reentrant structures are self-referencing; hence no cell in them will have a count of zero, even though the entire structure is unreachable. A modification of standard reference counting can be used to manaage the deallocation ...

15 An energy efficient garbage collector for java embedded devices
Paul Griffin, Witawas Srisa-an, J. Morris Chang
June 2005 **ACM SIGPLAN Notices , Proceedings of the 2005 ACM SIGPLAN/SIGBED conference on Languages, compilers, and tools for embedded systems LCTES'05**, Volume 40 Issue 7

Full text available: pdf(285.27 KB) Additional Information: full citation, abstract, references, index terms

This paper presents a detailed design and implementation of a power-efficient garbage collector for Java embedded systems. The proposed scheme is a hybrid between the standard mark-sweep-compact collector available in Sun's KVM and a limited-field reference counter. There are three benefits resulting from the proposed scheme. (a) the proposed scheme reclaims memory more efficiently and this results in less mark-sweep garbage collection invocations, (b) reduction in garbage collection invocations ...

Keywords: embedded systems, garbage collection, java mobile computing, virtual machines

16 A real-time garbage collector based on the lifetimes of objects

Henry Lieberman, Carl Hewitt
June 1983 **Communications of the ACM**, Volume 26 Issue 6

Full text available: pdf(1.37 MB) Additional Information: full citation, abstract, references, citings, index terms

In previous heap storage systems, the cost of creating objects and garbage collection is independent of the lifetime of the object. Since objects with short lifetimes account for a large portion of storage use, it is worth optimizing a garbage collector to reclaim storage for these objects more quickly. The garbage collector should spend proportionately less effort reclaiming objects with longer lifetimes. We present a garbage collection algorithm that (1) makes storage for short-li ...

Keywords: LISP, algorithms, languages, lisp, object-oriented programming, parallel processing, performance, real-time garbage collection, reference counting, virtual memory

17 Garbage collecting the Internet: a survey of distributed garbage collection

Saleh E. Abdullahi, Graem A. Ringwood
September 1998 **ACM Computing Surveys (CSUR)**, Volume 30 Issue 3

Full text available: pdf(337.65 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Internet programming languages such as Java present new challenges to garbage-collection design. The spectrum of garbage-collection schema for linked structures distributed over a network are reviewed here. Distributed garbage collectors are classified first because they evolved from single-address-space collectors. This taxonomy is used as a framework to explore distribution issues: locality of action, communication overhead and indeterministic communication latency.

Keywords: automatic storage reclamation, distributed, distributed file systems, distributed memories, distributed object-oriented management, memory management, network communication, object-oriented databases, reference counting

18 A unified theory of garbage collection

David F. Bacon, Perry Cheng, V. T. Rajan
October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN Conference on Object-oriented programming, systems, languages, and applications**, Volume 39 Issue 10

Full text available: pdf(223.52 KB) Additional Information: full citation, abstract, references, index terms

Tracing and reference counting are uniformly viewed as being fundamentally different approaches to garbage collection that possess very distinct performance properties. We have implemented high-performance collectors of both types, and in the process observed that the more we optimized them, the more similarly they behaved - that they seem to share some deep structure.

We present a formulation of the two algorithms that shows that they are in fact duals of each other. Intuitively, the ...

Keywords: graph algorithms, mark-and-sweep, reference counting, tracing

19 Concurrent garbage collection using program slices on multithreaded processors
Manoj Plakal, Charles N. Fischer
October 2000 **ACM SIGPLAN Notices , Proceedings of the 2nd international symposium on Memory management**, Volume 36 Issue 1
Full text available: pdf(957.62 KB) Additional Information: full citation, abstract, citings, index terms

We investigate reference counting in the context of a multi-threaded architecture by exploiting two observations: (1) reference-counting can be performed by a transformed program slice of the mutator that isolates heap references, and (2) hardware trends indicate that microprocessors in the near future will be able to execute multiple concurrent threads on a single chip. We generate a reference-counting collector as a transformed program slice of an application and then execute this slice in ...

20 Embedded systems: applications, solutions and techniques (EMBS): On designing a low-power garbage collector for java embedded devices: a case study
Paul Griffin, Witawas Srisa-An, J. Morris Chang
March 2005 **Proceedings of the 2005 ACM symposium on Applied computing**
Full text available: pdf(90.66 KB) Additional Information: full citation, abstract, references

This paper presents an energy consumption comparison between two well-known garbage collection algorithms---mark-sweep-compact and reference counting. Our goal is to evaluate the suitability of reference counting as an algorithm for memory-constrained Java embedded devices. We hypothesize that reference counting would be suitable because it has higher data locality, which should reduce the number of data-cache misses, and so reduce energy consumption. However, this benefit could be offset by the ...

Keywords: Java, embedded systems, garbage collection, power awareness

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Web Results **1** - **10** of about **549,000** for **reference counting collector**. (**0.18** seconds)

Why Garbage **Collection**?
Reference counting was an early garbage **collection** strategy. ... An advantage of this approach is that a **reference counting collector** can run in small ...
www.artima.com/insidejvm/ed2/gc3.html - 11k - Cached - Similar pages - Remove result

Integrating Generations with Advanced **Reference** (ResearchIndex)
5.5%: An on-the-fly **Reference Counting** Garbage **Collector** for Java - Levanoni, ... 11 Concurrent Cycle **Collection** in **Reference Counted** Systems - Bacon, ...
citeseer.ist.psu.edu/648792.html - 24k - Cached - Similar pages - Remove result

Ulterior **Reference Counting**: (ResearchIndex)
3.1: Ulterior **Reference Counting**: - Fast Garbage **Collection** (2003) (Correct) ... 9 fly **reference counting** garbage **collector** for Java - Levanoni, ...
citeseer.ist.psu.edu/636731.html - 22k - Cached - Similar pages - Remove result
[More results from citeseer.ist.psu.edu]

Java's garbage-collected heap
Reference counting garbage **collectors** distinguish live objects from garbage objects by ... **Reference counting** was an early garbage **collection** strategy; ...
www.javaworld.com/javaworld/jw-08-1996/jw-08-gc.html - Similar pages - Remove result

Reference Counting, Garbage **Collection**, and Object Lifetime ...
Unlike COM, the common language runtime does not use **reference counting** to govern object lifetime. Instead, the garbage **collector** traces object **references** ...
msdn.microsoft.com/library/en-us/vbcon/ html/vbconReferenceCountingGarbageCollectionObjectLifetime.asp - 13k - Cached - Similar pages - Remove result

Reference counting - Wikipedia, the free encyclopedia
Reference counting is often known as a garbage **collection** algorithm where ... Bacon describes a cycle-**collection** algorithm for **reference counting** systems ...
en.wikipedia.org/wiki/Reference_**counting** - 28k - Cached - Similar pages - Remove result

Garbage **collection** (computer science) - Wikipedia, the free ...
In contrast to tracing garbage **collection**, **reference counting** is a form of automatic memory management where each object has a **count** of the number of ...
en.wikipedia.org/wiki/Garbage_ collection_(computer_science) - 36k - Cached - Similar pages - Remove result

[PS] Integrating Generations with Advanced **Reference Counting** Garbage ...
File Format: Adobe PostScript - View as Text
It is used in most modern **reference counting collectors**. In particular, this method was ... All previous **reference counting collectors** execute 2n updates ...
www.cs.technion.ac.il/~erez/Papers/AzaPet-TR.ps - Similar pages - Remove result

Erez Petrank
An On-the-fly **Reference Counting** Garbage **Collector** for Java. ... Typically, mark-sweep and **reference-counting collectors** are susceptible to fragmentation. ...

www.cs.technion.ac.il/~erez/projects.html - 30k - Cached - Similar pages - Remove result

David Chappell :: Articles :: The Joy of **Reference Counting**
COM also doesn't necessarily assume the presence of an automatic garbage **collector**.
Instead, COM relies on **reference counting** to determine when an object ...
www.davidchappell.com/articles/ article_Refer_Counting.html - 23k - Cached - Similar pages - Remove result

Gooooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

reference counting collector Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google